Unix operating system. A user might create sub-groups in an object hierarchy under the top-level UNIX group for each category of resources, such as Unix volumes, Unix hosts, Unix applications and the like. When embodiments of the invention render or display specific resources within these groups, the naming scheme briefly outlined above of using the simple name or the simple name and the home for the object apply so that the user can quickly and intuitively understand which Unix resources are being referenced.

Embodiments of the invention support different types of group objects such as transparent group objects and terminal group objects. In particular, a user can create a group object to represent a group of objects sharing a common relationship and can make that group object a transparent group object. Thereafter, any objects initially created within (i.e., under or below that group object in the object hierarchy) that group obtain (i.e., are assigned) a home that is not equal to the transparent group object under which the new objects are added as child objects, but rather, the home for the new objects is set to the first non-transparent home object above the transparent object in the object hierarchy. In this manner, users can create groups for visual organization of representations of resources on a graphical user interface without effecting home relationships that need to be maintained. In other words, transparent objects provide a mechanism to create visual hierarchical relationships without effecting the name or object identifier of an object.

In addition to transparent objects, embodiments of the invention support a construct called a "terminal" object. If a user sets a group object to be terminal, then actions performed on that group object will not be carried out on members of the group. In other words, objects that are related to the terminal group object (i.e., that exist below the terminal group object in the object hierarchy) will not be effected by actions carried out by a user or process on the terminal object. As an example, a group of storage volumes might need to be represented by a terminal group object under a particular representation of a department object in order to convey that data in those storage volumes is related to a certain project in that department. However, if the terminal group object representing those volumes (the volume objects being below the group object in the object hierarchy) is acted upon under the department object (e.g., is deleted), the objects related under that group (i.e., the volumes in this example) will not also be deleted.

More specific details of the configuration and operation of embodiments of the invention will now be explained with reference Figures 1 through 7.

Figure 1 illustrates an example of a computing systems environment 100 that is suitable for use in explaining the operation of example embodiments of the invention. The computing system environment 100 in this example is a storage area network that includes a network medium 101 such as a high-speed data communications medium (e.g., Ethernet, optical network, or other type of network) which interconnects a plurality of resources 102 through 110 and allows the resources to communicate with each other. The resources include a variety of data storage systems 102-1 through 102-N, storage area network switches 103-1 through 103-M, host computer systems 104-1 through 104-P, client computer systems 105-1 through 105-S, and a computer system 110 configured in this example as a storage area network management station that is operated by a network manager 108 (i.e., a person responsible for managing the resources within the storage area network 100). While not shown in detail in this figure, certain of the resources may also include or operate one or more internal devices as well as software applications or processes which operate within or in conjunction with the illustrated resources. Such other components are to be considered resources as well for purposes of this explanation.

Of particular interest to this discussion, the management station computer system 110 operates a resource manager 121 configured according to embodiments of the invention. The resource manager 121 provides a graphical user interface 150 on a display 130 (e.g., a monitor or other display device) coupled to the computer system 110. The graphical user interface 150 includes a display of an example resource hierarchy including a plurality of representations of objects shown by object names arranged line by line in an indented or hierarchical manner. Each line in the object hierarchy correspond to a manageable resource or a group of such resources within the computing system environment 100. For example, the "ALL STORAGE SYSTEM" representation represents a group object in an object hierarchy under which objects are contained for each storage system 102-1 through 102-N.

Figure 2 illustrates a flow chart of the high-level processing steps and operations which preferred embodiments of the invention perform to create representations of objects, for example, such as shown in the example graphical user interface 150 in Figure 1. Preferably, the resource manager 121 operating within the computer system 110 performs the processing operations and steps shown in Figure 2. The processing in Figure 2 will be explained with respect to the creation of a representation of a single object. However, other embodiment can repetitively perform such operations to create multiple representations of multiple objects to form an object hierarchy that represents some or all of the resources in a computing system environment (e.g., 100).

In step 200, the resource manager 121 creates an object to represent a resource (e.g., one of 102 through 110) in a computing system environment (e.g., 100). Generally, the creation of an object can involve the instantiation of an object class (e.g., Java or C++ class) or other data structure which contains information related to the resource which the instantiated object represents. Different techniques for selecting or identifying resource(s) for which the resource manager 121 creates an object will be explained in more detail later. For purposes of this description, perhaps the user 108 of the computer system 110 provides an identification of a specific resource such as a volume of data storage which that user wants represented within the example graphical user interface 150 shown in Figure 1.

In step 201, the resource manager 121 assigns an object identifier to the object. The object identifier includes at least a simple name of the object as well as a home of the object. As briefly discussed above, the simple name of an object can be a user defined or automatically generated name, text or character string, or the like which this embodiment assigns to the object to identity the resource which that object represents. The home of the object is equivalent to a home location in an object hierarchy that identifies a home (e.g., a parent) object in the object hierarchy under which (either directly under or ancestrally under) the object is initially associated (i.e., created) as a child object. A home object thus defines a home context for the object. As such, the resource manager 121 assigns the home of the object to be the home location identifying the home object for that object in the object hierarchy. Assume for this example that an object hierarchy